

Improving the usability of scientific software with participatory design: a new interface design for radio astronomy visualisation software

Laurisha Rampersad

Department of Computer Science, University of Cape Town,
Town,
Private Bag X3, Rondebosch
Cape Town, South Africa 7701
laurisha.rampersad@gmail.com

Ed Elson

Department of Astronomy, University of Cape Town,
Private Bag X3, Rondebosch
Cape Town, South Africa 7701
elson.e.c@gmail.com

Sarah Blyth

Department of Astronomy, University of Cape Town,
Private Bag X3, Rondebosch
Cape Town, South Africa 7701
sarblyth@gmail.com

Michelle M. Kuttel

Department of Computer Science, University of Cape Town,
Private Bag X3, Rondebosch
Cape Town, South Africa 7701
mkuttel@cs.uct.ac.za

ABSTRACT

The importance of usability considerations in software development is well recognised. However, typically usability is not considered as an explicit goal in the development of scientific software, which is often done by developers with domain-specific knowledge but little formal software development training. In interactive software developed for international collaborations such as the Square Kilometre Array, usability is increasingly important. A possible solution is persistent collaboration between software developers and domain experts to design effective user interfaces. Here we carry out a User Centred participatory design approach to designing an astronomy visualisation interface. The methodology is iterative: in each iteration, a prototype interface was designed and then evaluated by users. Frequent consultation with domain experts produced an innovative design for an astronomy visualisation interface with improved usability.

CCS CONCEPTS

• **Human-centered computing** → **Empirical studies in interaction design**; • **Applied computing** → *Astronomy*;

KEYWORDS

usability, User Centred Design, participatory design, software interface, scientific software

ACM Reference format:

Laurisha Rampersad, Sarah Blyth, Ed Elson, and Michelle M. Kuttel. 2017. Improving the usability of scientific software with participatory design: a new interface design for radio astronomy visualisation software. In *Proceedings of SAICSIT '17, Thaba Nchu, South Africa, September 26–28, 2017*, 9 pages. <https://doi.org/10.1145/3129416.3129899>

1 INTRODUCTION

Modern scientific research is heavily dependent on software tools, tools which are often developed by scientific domain experts with little formal training in software engineering [2, 19, 21, 29]. As a result, many of these tools have poor usability: they are difficult to use, or involve complex workflows patched together from available software [19, 21]. Usability, encompassing utility, efficiency, effectiveness, learnability and memorability, is a key quality attribute of successful software systems [32]. However, where end users are fellow scientists working in the same field, domain experts who develop software often omit usability evaluation and user documentation entirely [17].

A User Centred Design (UCD) process, which focuses on the goals and needs of the system end-users, has been demonstrated to improve the usability of scientific software and can lead to an increase in the number users [20, 21]. For example, the Omero/Usable Image Project [21] used a team of usability experts and software engineers to build software for visualizing, managing, analysing, and annotating microscope images and metadata. Their participatory software development process, involving collaborative design workshops and iterative user evaluation, exposed many usability flaws: from a labelling problem in the search interface to a substantial hierarchy issue. Further, a project for the Hospital Italiano de Buenos Aires in Argentina redesigned the drug-drug interaction alert interface for an electronic health record system using UCD and participatory design techniques, producing a more effective and efficient interface [20]. However, despite the clear advantages, UCD

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SAICSIT '17, September 26–28, 2017, Thaba Nchu, South Africa

© 2017 Copyright held by the owner/author(s). Publication rights licensed to Association for Computing Machinery.

ACM ISBN 978-1-4503-5250-5/17/09...\$15.00

<https://doi.org/10.1145/3129416.3129899>

methods have not yet been broadly applied in scientific software development.

The Square Kilometre Array (SKA) project is an international venture involving 17 countries to build a radio telescope with an effective one million square meters of collection area [12]. This instrument will be physically located primarily in South Africa and Australia. The SKA will require a wide range of software tools able to handle the large data sizes and to be used by a very broad range of users world wide. In particular, there is a necessity for new visualization tools [25]. Visualization software is integral to research in radio astronomy as it enables exploration and analysis of the spectral data cubes produced by radio telescopes [11, 13]. A radio data cube is a 3D data structure that stores information about a specific volume of the universe obtained from a radio telescope. These cubes typically consist of a series of spatial images (Right Ascension versus Declination) stacked along a frequency axis to form a 3D voxel volume. Such spectral line cubes currently can be several gigabytes in size and the SKA is expected to produce data files up to a petabyte in size [14].

There are a number of tools available that allow astronomers to explore and analyse data cubes, including Karma [10], Visivo [3] and Gypsy [28]. These tools share fundamental features necessary for astronomers: exploring the data cubes visually by stepping through the cube 'slice-by-slice' (along the frequency axis channel by channel), zooming into and panning over a slice and the ability to obtain specific information on points in the data cube, such as coordinates and flux values. Some tools allow for 3D visualization of the entire data cube, enabling the user to visualize specific volumes in the data and rotate and zoom into certain areas of the data cube. 3D visualization is increasingly seen as a very desirable feature for cube visualization [25].

However, to date there has been little focus on optimising the software user interfaces of these tools. As a result, currently extant visualization tools, while showing good utility, have poor usability - they are not reportedly particularly efficient, effective or learnable. Indeed, the poor usability of much visualisation software for astronomy is well recognised: the Virtual Astronomy Observatory Science Council has emphasised the need for improved usability in software, especially in user interfaces [8]. It is also acknowledged that good usability is necessary for broad adoption of newly developed software by the astronomy community [6, 14].

In this work, we use UCD methods to design a new user interface for astronomy visualisation software with good usability, while accommodating the functionality of a typical radio astronomy visualisation tool such as Karma [10]. Our design methodology employs both adherence to industry-recommended design guidelines for interfaces [24] and a UCD participatory design process, with requirements gathering followed by three prototype design-and-evaluation iterations, with prototype increasing in fidelity with each iteration. We follow a participatory design process; our team includes two expert users as active participants in the process of requirements gathering, design and evaluation. Our final product is an interactive horizontal prototype of the interface that provides a broad view of the functionality, focusing on user interaction.

1.1 Contribution

This work describes the effective application of a participatory UCD methodology to improve the usability of a scientific software interface. We demonstrate the application of a range of usability evaluation methods in three successive design iterations of increasing fidelity. This process revealed specifications for functionality not currently supported in visualisation software for astronomy. The final artefact is a horizontal prototype, representing an improved design for the user interface to a generalised a radio astronomy visualisation software tool.

2 METHODOLOGY

We adapted the continuous user testing model as described in the Omero Project [21] for this work. Usability evaluation is best done continuously through interaction with end-users from the start to the end of a software project [7]. Our process began with a requirements gathering meeting with the expert users to determine the usability problems with current astronomy visualisation software and the needs of a typical astronomer when using a visualisation tool. We then developed a design for a new interface to an astronomy visualisation tool through three cycles of prototype design and expert user evaluation, with an emphasis on qualitative evaluation appropriate at the design stage. As the iterations progressed, the fidelity of the design prototypes increased, from a paper prototype to an interactive horizontal prototype. The two expert end users were identified to guide this project and, due to their deep involvement in the work, they are co-authors on this paper.

2.1 Design Principles

Adherence to lists of basic design principles is increasingly recommended for the development of scientific software [19]. However, for astronomy visualisation tools it is ironic that software designed for visualising data is often lacking in adherence to basic visualisation principles. These principles include mantras such as *overview first, zoom, filter, details on demand* [30]. To guide each design phase, we followed the Nielsen's ten usability heuristics for user interface design, as summarised in Table 1 [24]. Adherence to simple principles such as providing feedback for user actions has been demonstrated to greatly improve software usability. For example, the heuristics tabled by Nielsen are seemingly obvious but have been shown to collectively account for up to 95% of usability problems during evaluations of a system [24]. Simply enabling users to point and click rather than needing to type or remember information was found to remove 22% of all serious usability problems assessed.

2.2 Requirements Gathering

The goal of the requirements gathering phase was to identify the chief concerns and needs of a typical user of astronomy visualisation software. In order to gain a contextual understanding of how a visualization tool is used, requirements gathering was done in the expert users' standard work environment (a shared office setting) while the experts used the KARMA [10] standalone visualisation tool to explore a data cube. During the requirements gathering, a set of typical user tasks using were identified, while notes on the

Principle	Description
System Status	The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
Match between system and the real world	The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms.
User control and freedom	Support undo and redo functions.
Consistency and standards	Actions must be consistent both within the system and with platform conventions.
Error prevention	Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.
Recognition rather than recall	Minimize the user's memory load by making objects, actions, and options visible.
Flexibility and efficiency of use	Users should be able to tailor frequent actions.
Aesthetic and minimalist design	Dialogues should not contain information which is irrelevant or rarely needed.
Error messages	Error messages should be expressed in plain language, precisely indicate the problem, and constructively suggest a solution.
Help	Provide help and documentation.

Table 1: Nielsen's ten usability heuristics for user interface design [24].

usability of the software were taken along with user comments and suggestions.

2.3 Design Iterations

2.3.1 Iteration 1. In order to facilitate and encourage user criticism, in this first stage we produced two alternative paper prototype designs of a new astronomy visualization interface for the users to compare. Paper prototyping is advantageous as it allows for rapid production of multiple designs that are flexible and disposable. Paper prototyping also encourages the end users and designers to see the design as temporary and thus easy to change in terms of the functional aspects and task flow, instead of resisting changes [26]. It also avoids users comments on design aspects irrelevant to the early design stages, such as font or colour, and instead aims their focus on work flow and general layout.

The prototypes were evaluated via a Cognitive Walkthrough method [23]. Cognitive walkthroughs set up a task scenario for a user, who then talks through all the actions necessary to complete the task. The goal is to determine the difference between user expectations (and current level of understanding) and the reality of the interface through exploration. Cognitive walkthroughs with domain experts has been found to be effective for the identification of critical usability problems [9]. Our Cognitive Walkthrough evaluation session took place in a controlled office environment with the two expert users. The prototypes were described to the users in terms of tasks and features. The expert users then verbally reviewed and compared the two paper prototypes; comments and suggestions were recorded and occasionally drawn onto the prototypes or additional paper. This free-format interview allowed further brainstorming on design options to take place at this stage.

2.3.2 Iteration 2. The User Experience prototyping software Indigo Studio (by Infragistics) [15] was used during the second iteration for the higher fidelity prototype necessary to convey the complex workings of the interface. This software tool allowed for fast prototyping of a functional interface and recording user testing

metrics such as correct and incorrect clicks, as well as adding user comments.

Evaluations of this phase involved a task-based online user test, conducted using the website IndigoDesigned, which is affiliated with Indigo Studio. Three users were selected for this task; the two expert users involved in the participatory design process and a third selected through convenience sampling. During the test, users completed a set of tasks on the prototype, while all their movements, clicks and interactions were recorded. Users could also add comments to suggest improvements. This user test aimed to test whether or not the design was intuitive: test feedback indicated whether users clicked on the correct buttons or looked in the correct places to achieve a task.

2.3.3 Iteration 3. The third iteration of the prototype was also designed with Indigo Studio as an interactive horizontal prototype [27]. Horizontal prototypes demonstrate high level functionality across the prototype while avoiding lower level (in-depth) details [27].

Evaluation was conducted with the same three users as in *Iteration 2*. Three methods of evaluation were used for this final stage: a task-based user test, a System Usability Scale survey [5] and a final semi-structured interview. User Testing is an empirical test of the software. The tests take place either under real-world or controlled settings, with the test possibly being recorded on video or using computer logs. Users have to complete a set of tasks using the software with little assistance from the test conductors, after they have received the appropriate level of training for using the system. Jeffries et al.[23] found that user testing often finds the most critical problems in the software. The task-based user test aimed to evaluate whether the prototype fulfils user needs in terms of functionality, and also tested the overall usability of the prototype. It was conducted in office spaces with minimal interference. Users attempted to complete the tasks in the order assigned to them. These tasks ranged from simple interactions to those requiring multiple steps. The users were encouraged to Think Out Loud and were recorded talking through the process of completing the tasks [18].

A Think Out Loud evaluation requires the user to perform tasks using the prototype while talking the evaluators through these tasks; observations are recorded [18]. This method tests whether the prototype is fulfilling user needs in terms of tasks, as well as the flow and usability of the prototype.

The ten question System Usability Scale (SUS) Survey asks participants to rank aspects of the software on a scale from one to five and provides a simple method for calculating a quantitative score representing the user's impression of the overall usability of the system [5]. The SUS Survey is a robust and reliable measure of the usability of a system [4]: a SUS score of 75% indicates "good" usability and scores over 90% indicate truly superior products [1].

Finally, in the semi-structured interview, users commented in a conversational setting on their overall experience with the prototype in relation to the current software available.

3 RESULTS

3.1 Requirements Gathering

The requirements gathering stage produced the following model of an average astronomer working in radio astronomy interacting with visualisation software to examine radio data cubes. A researcher will use a visualisation tool regularly (once a day) to frequently (10-20 times a day) to view and explore different 3D data cubes. She will explore and analyse radio data cubes typically in 2D slice views across one of the three different axes: x,y, and z - typically Right Ascension, Declination, Frequency. A view of a cube will display annotations such as coordinate information and axes, which may be turned off. The researcher may view slices sequentially or else traverse a slice to focus on specific points of interest. In addition to this, the researcher may examine various overall statistical measures of the data, such as a histogram of pixel values integrated over a specific sub-volume.

The stand-alone visualization tools for astronomy currently available include KARMA [10], SAOImage DS9 [16] and VisIVO [3]. Other viewers are embedded in reduction and analysis packages, such as GIPSY [28, 31] and CASA [22]. Most of these software packages focus on a window interface for 2D visualization of slices of the 3D data cube; in some cases limited 3-D rendering is also present.

Many of the visualization tools reportedly have low usability [8], as user interfaces and interaction have not been the focus of development efforts. Interfaces typically have simplistic GUIs which require multiple fields and buttons to be addressed in order to complete a task. For example, Groningen Image Processing System (GIPSY) was first developed in 1992 to visualise astronomy data [31] and has been extended to 3D visualization [28]. Because of its long history, GIPSY is a powerful tool with a broad range of functions. However, the user interface is in need of improvement: it is highly technical and visually overwhelming, making it difficult for non-expert users to use, a fact acknowledged by the developers [28]. Another example is the module KARMA toolkit [10] which has many widgets to perform different functions. This greatly extends the usefulness of the toolkit and library. However, having several open window control panels adds to the cognitive complexity of tasks and lowers the usability of the system.

The Karma tool was chosen as the focus of analysis of existing tools, as it is frequently used by our expert users and representative of visualization tools overall. Our expert users expressed satisfaction with the functionality of Karma, but not its usability. They identified numerous shortfalls in terms of established usability principles.

For example, Karma shows a poor *match between system and the real world*: the interface is highly technical. When considering the principle of *flexibility and efficiency of use*, user preferences (colour maps, scaling, axes) in Karma must be respecified every time the program is booted, as these adjustments are not saved. Multiple windows must be opened and steps performed before basic tasks can be accomplished. Further, in terms of *error prevention*, in Karma closing one of these sub-windows exits the entire session, which can be very frustrating for the user. In terms of *aesthetic and minimalist design*, sub-menus in Karma are displayed as separate windows and this tends to clutter the workspace.

After discussing current problems, experts brainstormed ideal features for a visualisation interface with an overall improvement in interface usability.

Suggestions of the expert users can be broadly categorised into improved *navigation* through the data and improved *customisation* of the interface to suit a researcher's specific needs. In addition, we also identified desirable innovative functions that are not currently supported by existing software tools. The users reported that these functions would be useful as embedded features in visualisation software, as scientists frequently compute these queries manually (using, for example, hand-written notes).

Suggestions for navigation included viewing two radio cubes simultaneously for comparison; easy access to playing through 2D slices of the data, annotations of specific points in the data, colour scales to represent quantitative meaning and displaying co-ordinate data when hovering over the frame. Suggestions for improved customisation included the use of shortcut keys and standard sub-menus instead of new pop-up windows. Desirable additional functions identified by the users were: the ability to flag or annotate points in the data, the ability to capture images of the data and the ability to compare two data cubes. Note that these suggestions are not simply interface issues, but would extend the functionality of existing tools. We attempted to address as many of these requirements as possible in our first design.

3.2 Iteration 1

The first design cycle produced two paper prototypes depicting alternate interface options for the astronomy visualisation software (Figure 1a and b). Both designs provide an overview of the data in the main view, with further detail on demand. The first design focuses on navigation through the data cube, maximizing the screen space for the 3D data cube view in a single window, while the second design focuses on customisation of the features presented in the interface, for more efficient user workflow.

The designs have many features in common. Both use icons to represent functions to facilitate user recall. There are eight main functions which are consistent across both prototypes. The first function, represented by a camera, is to capture an image of the data and save it in a format suitable for publication. A flag icon represents

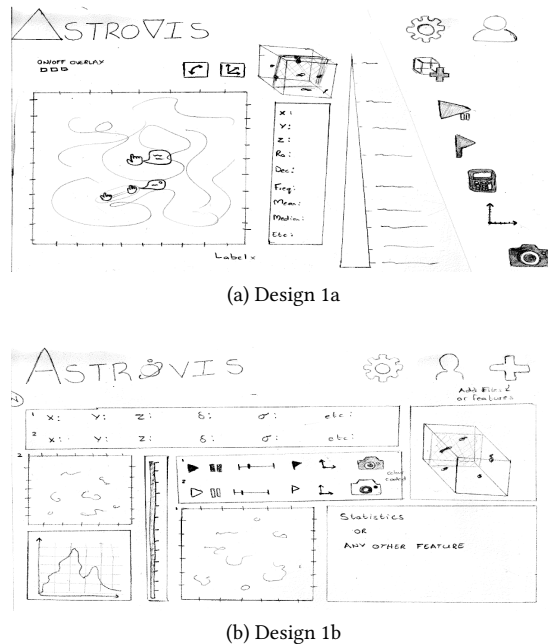


Figure 1: Home view of the two paper prototype designs produced and evaluated in iteration 1.

the ability to flag (annotate) points in data with comments. Once clicked, the cursor will change to a pointer to allowing the user to select a point on the frame to flag and to save a note or comment on that data. Users can swap axes (e.g. from XY to XZ) easily using the cross-shaped axes icon. A settings menu is represented by a gear icon to easily change and save preferences for future sessions. Most importantly, a user profile allows a user to save preferences in settings, as well as for saved captures and flags to be stored. The ability to play through consecutive 2D slides of the 3D cube in the main screen is represented by the familiar triangular "play" icon which changes to a pause icon when clicked and vice versa. The calculator icon represents a section where further calculations, such as statistical measures and histograms, can be found. Finally, a cube with a '+' on it represents the ability to split the screen in two and view a second data cube for comparison. An additional feature provides an interactive three dimensional cube of the data along with the images viewed by frame to enhance contextual understanding of the data.

While the layout is different, all functions, icons and buttons perform the same actions across both designs, which only differ significantly in the layout of the interface, as described below.

3.2.1 Design 1a. The first design (Figure 1a) has a large screen for display of the 3D data cube, to allow for as much space as possible for users to interrogate the data. An expanding side menu on the right contains all the icons representing additional functionality. This side menu allows for less clutter on screen and more space in the visual section of the layout. The play button (middle), allows slices of a radio cube in the main visualisation of the slices to be stepped through or played as a film. There is also an option to alternate the main focus of the window to an interactive view of

the 3D data cube and vice versa. To the immediate left of the side menu is a triangular scale depicting the values corresponding to the colour scale/map used in the main visualisation of the data cube. There is also a space dedicated to showing important updated data (co-ordinate and statistical) which is easy to view without referring to the menus.

3.2.2 Design 1b. The second design (Figure 1(b)) is a customisable widget-based view that aims to allow expert (high volume) users to set up their most commonly used features as fixtures on the main screen with as little effort as possible. Each box and object represents different functions and tools which can be dragged on- and off-screen to allow expert users to customise their own experience. Even though this is a customisable interface, the layout drawn is the standard recommended view which would be available the first time the program is used. Thereafter, users could change certain widgets/features to incorporate their most used functions and make them all easily accessible (e.g. some further calculations). The big plus icon in the top right of the screen will be used to add features to the screen. If a widget is dragged towards the plus button, it will turn into an 'x' and releasing the widget over this x removes it from the screen. Next to this plus button is the profile and settings, which will be permanently available in the header ribbon.

3.2.3 Evaluation. For Design 1a, the expert users appreciated the overview of the data cube, with details on demand [30]. The range of functions for navigation, including new functions such as flagging the data with comments, were welcomed, as was customisation with a user profile for saved settings, such as visible axes and data colour scales. Users remarked that viewing and interacting with data cubes will be much faster without all the set up usually required.

Some of the new features needed adjustment. Users said that screenshots of the data would not be of high enough resolution for publication. They suggested instead the option for saving a vector graphics or PNG image file. Furthermore, exported images could have tagged meta data (session information such as file name, time, and custom fields). Further adjustments included that the add button should implement a split screen for two cubes (for comparison) and tabs for three or more cubes; a file browser in the main screen to show files currently visualised and to find more; flags saved in a separate file to avoid altering the original data and to allow for future overlay of flags onto another data file (e.g. an optical file); and that editing the axes via the shortcut menu buttons should give the user the option to save these changes to the profile.

For Design 1b, the expert users said that the customisable widget-based view was potentially very useful and efficient. However, they thought that there might be a problem with adoption of such an unconventional interface; Design 1a is simpler and more likely to be adopted by more traditional users. The expert users suggested combining the two interfaces, with Design 1a as a home view and Design 1b as the expert view.

3.3 Iteration 2

The second design phase incorporated feedback from expert users to create an interactive simplified prototype of the GUI¹. The

¹The second design is available at <https://indigodesigned.com/share/3mn759zp0y06>

prototype used dummy data for the text, statistics and images that the user would encounter in a functional implementation. Features carried through from the first prototype have the same functionality in this iteration, unless otherwise specified.

3.3.1 Design. The second prototype, shown in Figure 2, combined both designs from the first iteration, according to suggestions from the expert users. The Home view layout (Figure 2a) is derived from Design 1a. The main area of the screen is devoted to the current slice view of the data cube, a 3D overview of the entire cube to enable navigation (with a play button and slider), a file browser and a statistics panel. All panels are resizable and customisable, as in Design 1b. A familiar triangular expanding "hamburger" menu houses the icon buttons, shown expanded in Figure 2b. The menu items have been reduced to six items from eight: the triangular "play" icon is now in the main screen and the options for axes is now in the settings submenu. The user profile option expands the triangular menu fully, to cover most of the screen (Figure 2c). It lists recently used files, image captures and flags added. Settings allows the user to change and save preferences (Figure 2d), including the presence of axes.

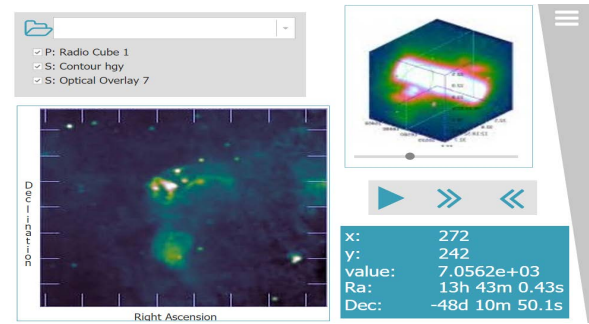
The capturing image function in this prototype includes export formats and metadata for files. The flagging function changes the cursor when selected. The 'Further Calculations' function (calculator icon) is a placeholder for complex statistical data a user might need. The add button now allows the user to: add data to the home screen (either as an additional viewing panel or as an overlay of data onto the primary radio data cube); add other features (such as statistics or a file browser) to the home screen as additional panels; and remove existing panels using checkboxes. This is intended to aid the customisation of the screen view.

3.3.2 Evaluation. Evaluation showed a high level of user satisfaction with the second prototype: overall the functionality was described as useful and an improvement in the usability of astronomy software.

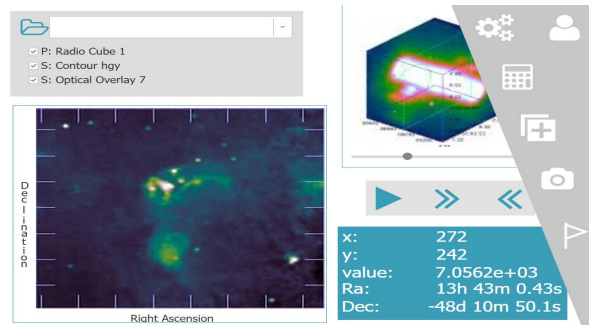
In terms of navigation, users particularly liked the 3D cube, as the 3D view provides recognition over recall: it gives overview and context for the standard slice view, especially when moving through frames with the play button and slider. In terms of flexibility and efficiency, users appreciated features readily available on the main view (without requiring menus to access) and were all enthusiastic about the stored user profile for customisable settings. Users found informative system status elements such as the statistics panel on the home screen and a list of recent captures and flags useful.

Users were particularly enthusiastic about the ease of access to the additional features provided by the interface: the annotation/flagging option and the stored user profile. In addition, exporting images is much smoother than in existing packages. One user commented that the annotation feature and the user profile for customisable settings make this prototype a great improvement on current software.

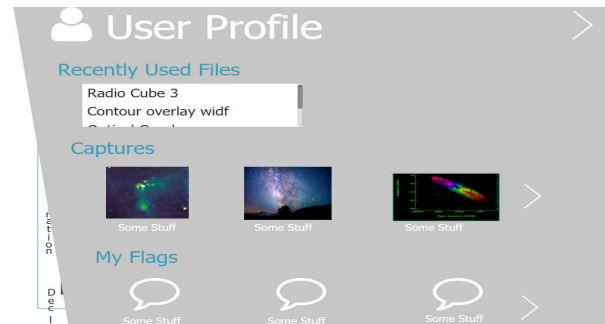
Users suggested some improvements to navigation and efficiency, such as making the flag option more immediately visible. Other suggestions addressed consistency and intuition: users suggested listing saved annotations and screen captures under their respective icons (flag and camera) rather than only in the profile. A user



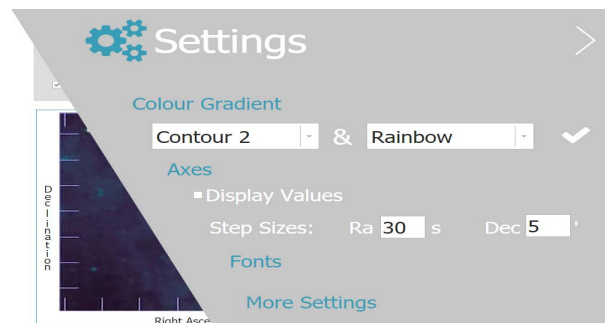
(a) Home View



(b) Expanded Menu



(c) Profile Expanded View



(d) Settings Expanded Menu

Figure 2: Prototype Design 2: overview.

further suggested that a separate menu house visualisation preferences such as colour maps, rather than their current home in settings, where they expected only global application preferences (such as font) to be listed. Further suggestions for improved flexibility and efficiency were to provide access to more tools from the home screen (such as editing contrast, swapping axes, colour maps, scaling schemes etc) and to include shortcuts for high volume users (such as ctrl+click to skip ten frames instead of skipping through one at a time).

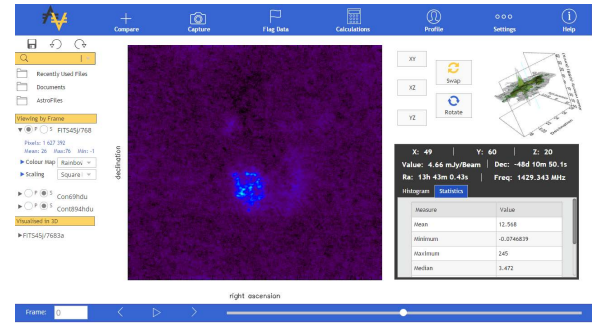
Finally, some user suggestions addressed the aesthetic and minimalist design: a user felt that the triangular toolbar takes up too much space in this prototype and suggested that these tools were rather permanently exposed, with perhaps an option to hide them. This addresses the usability concept of recognition rather than recall. Some feedback was conflicting, however, as one participant liked the icons - calling them intuitive - while another participant found some of the icons unintuitive and suggested leaving the labels on the icons all the time rather than only being visible when hovering over an icon.

3.4 Iteration 3

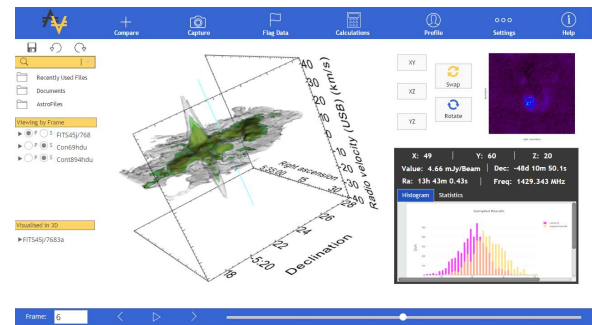
The final Horizontal-Prototype [27]² addresses all expert user feedback in the previous iterations. Our design goals referenced established usability heuristics [24, 30] in aiming to achieve an aesthetic and minimalist design that maintains consistency with conventional practices, while reducing the cognitive load on the user with icons for recognition and incorporating functions intuitively.

3.4.1 Design. Design 3 has the same functionality as previous iterations, maximising space for the data visualisation. Figure 3a depicts the initial view of the system. The main central panel by default contains the slice visualisation (showing the current frame of the data cube) and occupies the majority of the screen space. A small sub-panel on the right contains the 3D view of the current data cube, which may be rotated. To the right of the main panel is a control panel with five buttons. Users may interchange these two data visualisation panels using the 'swap' button in this control panel (Figure 3b). The 'rotate' button allows the user to rotate the 3D image. The other three buttons in this section – XY, XZ, YX – allow the user to swap the axes easily in the slice viewer. The toolbar along the bottom of the screen allows the user to skip to certain frames in the cube, to play through successive frames and to scroll to a particular frame using a slider.

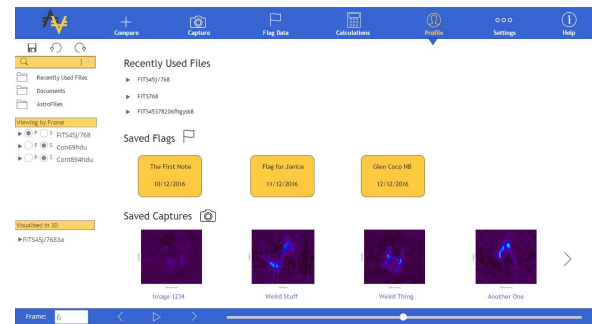
Functions are more directly displayed in this prototype, to allow for recognition, rather than recall. The bottom right of the screen holds an informational field with tabs to toggle between histograms and statistics of the data cube. A file browser is listed to the left of the main panel. Settings such as colour map and scaling, are listed below this, for easy access. Items listed here can be expanded by clicking on the arrow icons (consistent with current interface conventions) to allow for rapid changing of colour maps and scaling schemes. Finally, the triangular side menu in Design 2 is replaced by a more compact horizontal ribbon menu of functions at the top of the screen. This is used to access submenus, such as the profile submenu shown in Figure 3c. Annotations and image captures



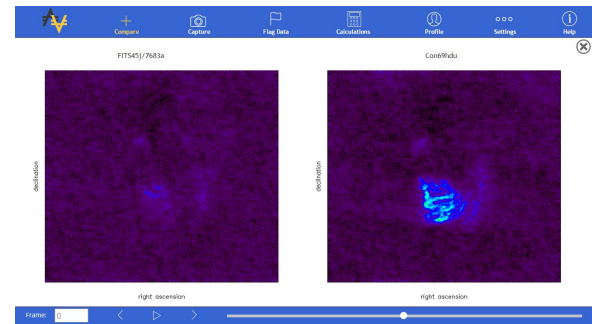
(a) Home View



(b) Switching Visualisation Panels



(c) Profile View



(d) Compare View

Figure 3: Final Prototype - Design 3.

² The prototype is available at <https://indigodesigned.com/share/vn61mym8xkc4>

are found under their respective menus. For flagging, the cursor becomes a pointer when selecting a point in the frame to flag. Once selected, a dialog prompts users to annotate and save the flag. The capture feature works similarly and can directly export and save captured images.

A new feature, *compare* allows users to select and compare two data sets Figure 3d. In this view, data sets can viewed by frame or played as a film simultaneously. The rest of the screen is removed for this feature in order to maximise the ability to explore and compare the data.

3.4.2 Evaluation. Feedback on the final design was largely positive: the SUS survey results for each expert user were 87.5%, 92.5% and 97.5%, indicating that all three users found the interface to have excellent usability [1].

Most tasks were completed with ease and icons were found to be intuitive. In terms of navigation, users particularly liked having both the 3D cube and 2D frame view in one interface. Users enjoyed the clean and modern aesthetic, which they felt was consistent with modern software. They also remarked that the clean interface made it easy to find relevant functions and they liked the listing of all the data functions on the screen. The additional features – annotations, image captures and data comparisons – were rated highly.

There was some confusion about the rotate function needing a click and hold interaction. One user found this unintuitive, another user liked it and the third user felt it was a matter of preference. Most users navigated the control panel (consisting of the XY, XZ, YZ, swap and rotate buttons) quickly. They found it clear and intuitive due to the icons and labelling. They also remarked that the directly accessible buttons would speed up data analysis. However, one user was confused by which visualisation each button refers to (either the 2D viewing by frame, 3D cube or both). This user suggested that the swapping axes buttons (XY, XZ, YZ) should be placed on the frame view and that the rotate button should be on the 3D cube with a rotation element to pop up.

Users had other suggestions including to: add clarity to the scientific analysis; add a description of the format type when saving images; use a dialog box with drop downs rather than submenus to capture images and to add keyboard shortcuts as astronomers tend to use these often. One user also noted that although the data was displayed in a useful way, there was not much indication as to how this prototype would handle very large data sets.

4 DISCUSSION

Overall, we found that our iterative User Centred Design approach involving participation with expert users was very effective for designing a new interface for an astronomy visualisation tool. Separating the interface design from the software implementation allowed for an emphatic focus on user needs, adherence to accepted usability heuristics, as well as greater freedom in the design process. Further, the involvement of domain experts in the design process was very effective in determining user requirements in a complex domain, as has been found in other studies of software development for science [21]. Iterative phases were useful for refining and clarifying usability requirements, allowing us at the design stage to avoid costly misunderstandings that would impact on the usability of a final implementation. However, it is important to note for

future projects that this methodology is limited by the scarcity of suitable local domain experts. In future work, methods of accessing evaluation by the international astronomy community would be beneficial.

In terms of the methodology, paper prototyping proved to be difficult, as typical visualisation software has complex interactions. However, paper sketches did allow for more innovative designs and the end users enjoyed the ability to manipulate the prototype and to point to different aspects when giving feedback. In addition, users could draw on the paper prototypes in the review sessions, allowing for co-designed elements. For higher fidelity, prototyping software was very effective, especially metrics provided by the on-line user test, such comments from the users. Interactive, computer based prototypes proved to be successful in examining usability and effectiveness. The SUS survey was very useful in providing a clear quantitative measure of each user's impression of the usability of the final prototype.

Our final design, which was rated highly for usability, differs noticeably from the interfaces of existing astronomy visualisation tools in terms of aesthetic and presentation of the functionality. There are a number of general aspects in the final design that can be utilised by other visualisation tools. Firstly, users appreciated having an overview of all data in the home screen (while maintaining a clean layout) and then further detail on demand. This finding is in accordance with Shneiderman's visualisation mantra; *overview first, zoom, filter, details on demand* [30].

Secondly, we found that icons in the interface reduced the need for recall, as users either found icons either immediately intuitive or easily learnable. Users were able to point and click rather than type and remember - which Nielson found solved up to 22% of usability problems [24]. This is important to recognise, as icons are rarely used in astronomy visualisation tools

Finally, we found that the expert users had needs that were currently unmet by existing software and have specified an effective design of desirable innovative functions for future software, allowing users to flag, capture and compare data.

We expect that our process of progressive, consultative refinement will result in more usable interface in a final software implementation. However, we have not assessed the feasibility of the final design. While many of the features are aesthetic or functional, some might be difficult to accomplish in practice, such as loading multiple data sets in different views (frame-by-frame and 3D). Despite this, it would be worth trying to adhere to our final specification as much as possible, as users found Design 3 to be an improvement on current software in terms of both usability and useful features. In the words of one participant: *"Overall I think this interface is far superior to some of the existing visualisation software commonly used in radio astronomy in terms of usability (i.e. easy to pick up and use without training) and in terms of usefulness."*

5 CONCLUSIONS

We employed an iterative User Centred Design approach involving participation with expert users to design an interface to an astronomy visualisation tool with improved usability over current software. The SUS survey indicates that users found the usability of the final design to be "excellent". Future work can build a fully

functional implementation based on this design or, alternatively, extract the design elements used and adapt them to other scientific software projects. Our general recommendations for interfaces to astronomy visualisation tools are to reduce cognitive load with icons and provide an overview of the data first, with details on demand. We also provide specifications for functionality not currently supported in visualisation software for astronomy: allowing users to flag, capture and compare data. We recommend our collaborative approach of separate interface design for future software development efforts in astronomy. Given the clear benefits and low cost of our design process, we hope that future scientific software projects will incorporate UCD into the software development process for improved usability.

REFERENCES

- [1] Aaron Bangor, Philip T Kortum, and James T Miller. 2008. An empirical evaluation of the system usability scale. *Intl. Journal of Human-Computer Interaction* 24, 6 (2008), 574–594.
- [2] Susan M Baxter, Steven W Day, Jacquelyn S Fetrow, and Stephanie J Reisinger. 2006. Scientific Software Development Is Not an Oxymoron. *PLoS Comput Biol* 2, 9 (2006), e87.
- [3] U Becciani, A Costa, V Antonuccio-Delogu, G Caniglia, M Comparato, C Gheller, Z Jin, Mel Krokos, and P Massimino. 2010. VisIVO-integrated tools and services for large-scale astrophysical visualization. *Publications of the Astronomical Society of the Pacific* 122, 887 (2010), 119.
- [4] J. Brooke. 2013. SUS: a retrospective. *Journal of Usability Studies* 8, 2 (2013), 29–40.
- [5] John Brooke et al. 1996. SUS-A quick and dirty usability scale. *Usability evaluation in industry* 189, 194 (1996), 4–7.
- [6] G Chiozzi, K Gillies, B Goodrich, S Wampler, J Johnson, K McCann, G Schumacher, and D Silva. 2007. Trends in software for large astronomy projects. In *11th ICALEPCS Int. Conf. on Accelerator & Large Experimental Physics Control Systems*, Knoxville. 13–17.
- [7] Joseph S Dumas and Janice Redish. 1999. *A Practical Guide to Usability Testing*. Intellect Books.
- [8] G. Fabbiano, D. Calzetti, C. Carilli, G. Djorgovski, P. Eskridge, Z. Ivezic, E. Feigelson, A.; Goodman, B. Madore, S. SEAGER, A. SODERBERG, and T. RECTOR. 2010. Recommendations of the VAO-Science Council. (06/2010 2010), 9 pages. <http://arxiv.org/abs/1006.2168>
- [9] Asbjørn Følstad. 2007. Work-domain experts as evaluators: usability inspection of domain-specific work-support systems. *International Journal of Human-Computer Interaction* 22, 3 (2007), 217–245.
- [10] Richard Gooch. 1996. KARMA: a visualization test-bed. In *Astronomical Data Analysis Software and Systems V*, G.H. Jacoby and J. Barnes (Eds.). ASP Conf. Series, Vol. 101. Astron. Soc. Pac., 80–82.
- [11] Alyssa A Goodman. 2012. Principles of high-dimensional data visualization in astronomy. *Astronomische Nachrichten* 333, 5–6 (2012), 505–514.
- [12] PJ Hall. 2005. The Square Kilometre Array: An International Engineering Perspective. In *The Square Kilometre Array: An Engineering Perspective*. Springer, 5–16.
- [13] A.H. Hassan, C.J. Fluke, and D.G. Barnes. 2011. Interactive visualization of the largest radioastronomy cubes. *New Astronomy* 16, 2 (2011), 100 – 109. <https://doi.org/10.1016/j.newast.2010.07.009>
- [14] Amr Hassan and Christopher J Fluke. 2011. Scientific visualization in astronomy: Towards the petascale astronomy era. *Publications of the Astronomical Society of Australia* 28, 02 (2011), 150–170.
- [15] Infragisticc. 2016. Indigo Studio. (2016). <http://www.infragistics.com/products/indigo-studio> [Software].
- [16] WA Joye and E Mandel. 2003. New features of SAOImage DS9. In *Astronomical data analysis software and systems XII (Astronomical Society of the Pacific Conference Series)*, H. E. Payne, R. I. Jedrzejewski, and R. N. Hook (Eds.), Vol. 295. Astron. Soc. Pac., Astron. Soc. Pac., 489–490.
- [17] Diane Kelly and Rebecca Sanders. 2008. The challenge of testing scientific software. In *Proc. Conf. for the Association for Software Testing (CAST)*. Toronto, 30–36.
- [18] Clayton Lewis and John Rieman. 1993. *Task-centered user interface design. A practical introduction*. Clayton Lewis and John Rieman, Boulder, Colorado.
- [19] M. List, P. Ebert, and F. Albrecht. 2017. Ten Simple Rules for Developing Usable Software in Computational Biology. *PLoS Comput Biol* 13, 1 (2017), e1005265.
- [20] D. R. Luna, D. A. Rizzato Ledo, C. M. Otero, M. R. Risk, and F. G. B. de Quirós. 2017. User-centered design improves the usability of drug-drug interaction alerts: Experimental comparison of interfaces. *J. Biomed. Inform.* 66 (2017), 204–213.
- [21] Catriona Macaulay, David Sloan, Xinyi Jiang, Paula Forbes, Scott Loynton, Jason R Swedlow, and Peter Gregor. 2009. Usability and user-centered design in scientific software development. *IEEE Software* 26, 1 (2009), 96.
- [22] J.P. McMullin, B. Waters, D. Schiebe, W. Young, and K. Golap. 2007. CASA Architecture and applications. In *Astronomical Data Analysis Software and Systems XVI*, R.A. Shaw, F. Hill, and D.J. Bell (Eds.). Astron. Soc. Pac. Conf. Ser., Vol. 376. Astron. Soc. Pac., Tucson, Arizona, USA, 127.
- [23] James R Miller and Robin Jeffries. 1992. Interface-usability evaluation: science of trade-offs. *Software, IEEE* 9, 5 (1992), 97–98.
- [24] Jakob Nielsen. 1994. Enhancing the explanatory power of usability heuristics. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM, 152–158.
- [25] Davide Punzo, JM van der Hulst, JBTM Roerdink, TA Oosterloo, M Ramatsoku, and MAW Verheijen. 2015. The role of 3-D interactive visualization in blind surveys of HI in galaxies. *Astronomy and Computing* 12 (2015), 86–99.
- [26] Marc Rettig. 1994. Prototyping for tiny fingers. *Commun. ACM* 37, 4 (1994), 21–27.
- [27] Jim Rudd, Ken Stern, and Scott Isensee. 1996. Low vs. High-Fidelity Prototyping Debate. *interactions* 3, 1 (1996), 76–85.
- [28] JE Ruiz, JD Santander-Vela, V Espigares, L Verdes-Montenegro, and JM van der Hulst. 2009. GIPSY 3D: Analysis, Visualization and VO Tools for Datacubes. In *Astronomical Data Analysis Software and Systems XVIII*, Vol. 411. Astron. Soc. Pac., Tucson, Arizona, USA, 406.
- [29] Judith Segal and Chris Morris. 2008. Developing scientific software. *Software, IEEE* 25, 4 (2008), 18–20.
- [30] Ben Shneiderman. 1996. The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*. IEEE Computer Society, IEEE Computer Society, Washington, DC, USA, 336–343.
- [31] J. M. van der Hulst, J. P. Terlouw, K. G. Begeman, W. Zwitter, and P. R. Roelofsma. 1992. The Groningen Image Processing System, GIPSY. In *Astronomical Data Analysis Software and Systems I*, Diana M. Worrall, Chris Biemesderfer, and Jeannette Barnes (Eds.). A.S.P. Conference Series, Vol. 25. Astronomical Society of the Pacific, 131.
- [32] S. Winter, S. Wagner, and F. Deissenboeck. 2008. A Comprehensive Model of Usability. In *Engineering Interactive Systems*, J. Gulliksen, M.B. Harning, P. Palanque, G. C. van der Veer G.C., and J. Wesson (Eds.). Lecture Notes in Computer Science, Vol. 4940. Springer, Berlin, Heidelberg.